

ADANA SCIENCE AND TECHNOLOGY UNIVERSITY

Introduction to Computer Programming II

Objectives for today

- Examples with Pointers and arrays

Example

```
int fun(int* &p, int *q)
{
    *p=12;
    p=p+3;
    cout<<"p= "<<*p<<endl;
    q=p-1;
    *q=*(p+1) + *q;
    cout<<"p= "<<*p<<" *q = "<<*q<<endl;
    *p += 2;
    cout<<"p= "<<*p<<" *q = "<<*q<<endl;
    return (*p) % (*q);
}
int main ( )
{
    int y[5] = {1,2,3,4,5};
    int *p = y;
    int *q = new int;
    *q = fun(p,q);
    for (int i=0; i<5; i++)
        cout<<y[i]<<endl;
    cout<<"p= "<<*p<<" *q = "<<" *q"<<endl;
    return 0;
}
```

- What is the output of the program on the left?

Example

```
int fun(int* &p, int *q)
{
    *p=12;
    p=p+3;
    cout<<"p= "<<*p<<endl;
    q=p-1;
    *q=*(p+1) + *q;
    cout<<"p= "<<*p<<" *q = "<<*q<<endl;
    *p += 2;
    cout<<"p= "<<*p<<" *q = "<<*q<<endl;
    return (*p) % (*q);
}
```

```
int main ( )
{
    int y[5] = {1,2,3,4,5};
    int *p = y;
    int *q = new int;
    *q = fun(p,q);
    for (int i=0; i<5; i++)
        cout<<y[i]<<endl;
    cout<<"p= "<<*p<<" *q = "<<"
*q"<<endl;
    return 0;
}
```

Name of the variable	y[0]	y[1]	y[2]	y[3]	y[4]
Value of the variable	1	2	3	4	5
Address of the variable	1000	1001	1002	1003	1004

Example

```
int fun(int* &p, int *q)
{
    *p=12;
    p=p+3;
    cout<<"p= "<<*p<<endl;
    q=p-1;
    *q=*(p+1) + *q;
    cout<<"p= "<<*p<<" *q = "<<*q<<endl;
    *p += 2;
    cout<<"p= "<<*p<<" *q = "<<*q<<endl;
    return (*p) % (*q);
}
```

```
int main ( )
{
    int y[5] = {1,2,3,4,5};
    int *p = y;
    int *q = new int;
    *q = fun(p,q);
    for (int i=0; i<5; i++)
        cout<<y[i]<<endl;
    cout<<"p= "<<*p<<" *q = "<<"
*q"<<endl;
    return 0;
}
```

Name of the variable	y[0]	y[1]	y[2]	y[3]	y[4]	**p**
Value of the variable	1	2	3	4	5	**1000**
Address of the variable	1000	1001	1002	1003	1004	**2000**

Example

```
int fun(int* &p, int *q)
{
    *p=12;
    p=p+3;
    cout<<"p= "<<*p<<endl;
    q=p-1;
    *q=*(p+1) + *q;
    cout<<"p= "<<*p<<" *q = "<<*q<<endl;
    *p += 2;
    cout<<"p= "<<*p<<" *q = "<<*q<<endl;
    return (*p) % (*q);
}
```

```
int main ( )
{
    int y[5] = {1,2,3,4,5};
    int *p = y;
    int *q = new int;
    *q = fun(p,q);
    for (int i=0; i<5; i++)
        cout<<y[i]<<endl;
    cout<<"p= "<<*p<<" *q = "<<"
*q"<<endl;
    return 0;
}
```

Name of the variable	y[0]	y[1]	y[2]	y[3]	y[4]	p	**q**	
Value of the variable	1	2	3	4	5	1000	**4000**	
Address of the variable	1000	1001	1002	1003	1004	2000	**3000**	4000

Example

```
int fun(int* &p, int *q)
{
    *p=12;
    p=p+3;
    cout<<"p= "<<*p<<endl;
    q=p-1;
    *q=*(p+1) + *q;
    cout<<"p= "<<*p<<" *q = "<<*q<<endl;
    *p += 2;
    cout<<"p= "<<*p<<" *q = "<<*q<<endl;
    return (*p) % (*q);
}
```

```
int main ( )
{
    int y[5] = {1,2,3,4,5};
    int *p = y;
    int *q = new int;
    *q = fun(p,q);
    for (int i=0; i<5; i++)
        cout<<y[i]<<endl;
    cout<<"p= "<<*p<<" *q = "<<" *q"<<endl;
    return 0;
}
```

	main FUNCT.								**fun** FUNCT.	
Name of the variable	y[0]	y[1]	y[2]	y[3]	y[4]	p	q		**p&**	**q**
Value of the variable	1	2	3	4	5	1000	4000			**4000**
Address of the variable	1000	1001	1002	1003	1004	2000	3000	4000		**5000**

Example

```
int fun(int* &p, int *q)
{
    *p=12;
    p=p+3;
    cout<<"p= "<<*p<<endl;
    q=p-1;
    *q=*(p+1) + *q;
    cout<<"p= "<<*p<<" *q = "<<*q<<endl;
    *p += 2;
    cout<<"p= "<<*p<<" *q = "<<*q<<endl;
    return (*p) % (*q);
}
```

```
int main ( )
{
    int y[5] = {1,2,3,4,5};
    int *p = y;
    int *q = new int;
    *q = fun(p,q);
    for (int i=0; i<5; i++)
        cout<<y[i]<<endl;
    cout<<"p= "<<*p<<" *q = "<<" *q"<<endl;
    return 0;
}
```

PROGRAM OUTPUT

p= 4

p= 4 *q= 8

p= 6 *q= 8

Returning value = 6%8 = 6

12 8 6 1003 1002

	main FUNCT								fun FUNCT.	
Name of the variable	y[0]	y[1]	y[2]	y[3]	y[4]	p	q		**p&**	**q**
Value of the variable	1	2	3	4	5	1000	4000			4000
Address of the variable	1000	1001	1002	1003	1004	2000	3000	4000		5000

Example

```
int fun(int* &p, int *q)
{
    *p=12;
    p=p+3;
    cout<<"p= "<<*p<<endl;
    q=p-1;
    *q=*(p+1) + *q;
    cout<<"p= "<<*p<<" *q = "<<*q<<endl;
    *p += 2;
    cout<<"p= "<<*p<<" *q = "<<*q<<endl;
    return (*p) % (*q);
}
```

```
int main ( )
{
    int y[5] = {1,2,3,4,5};
    int *p = y;
    int *q = new int;
    *q = fun(p,q);
    for (int i=0; i<5; i++)
        cout<<y[i]<<endl;
    cout<<"p= "<<*p<<" *q = "<<" *q"<<endl;
    return 0;
}
```

<u>PROGRAM OUTPUT</u>

p= 4
p= 4 *q= 8
p= 6 *q= 8
12
2
8
6
5
p= 6 *q= *q

	main FUNCT.							
Name of the variable	y[0]	y[1]	y[2]	y[3]	y[4]	p	q	
Value of the variable	12	2	8	6	5	1003	4000	**6**
Address of the variable	1000	1001	1002	1003	1004	2000	3000	4000

Example

Example

```
void fun1 (int *p, int *q){
    *p = 100;
    p = p + 2;
    *p = *q-1;
    *q = *(p+2);
    cout << "p=" << *p << " q=" << *q << "\n";
}
bool fun2 (int x[], int size) {
    int *p = x;
    int *q;
   q = p;
    for (int i=1; i<size; i++)    {
        x[i] += x[i-1];
        cout << x[i] << " ";
    }
    cout << "\n";
    fun1 (p, q);
    return (*p == *q);
}
int main ( ){
    int a[5] = {1,5,3,2,20};
    if (fun2 (a,5))
        cout << "true\n";
    else
        cout << "false\n";
    for (int i=0; i<5; i++)
        cout << a[i] << " ";
    return 0;
}
```

- What is the output of the program on the left?

Example

```
void fun1 (int *p, int *q){
    *p = 100;
    p = p + 2;
    *p = *q-1;
    *q = *(p+2);
    cout << "p=" << *p << " q=" << *q << "\n";
}
bool fun2 (int x[], int size) {
    int *p = x;
    int *q;
   q = p;
    for (int i=1; i<size; i++)    {
        x[i] += x[i-1];
        cout << x[i] << " ";
    }
    cout << "\n";
    fun1 (p, q);
    return (*p == *q);
}
```

```
int main ( ){
    int a[5] = {1,5,3,2,20};
    if (fun2 (a,5))
        cout << "true\n";
    else
        cout << "false\n";
    for (int i=0; i<5; i++)
        cout << a[i] << " ";
    return 0;
}
```

Name	a[0]	a[1]	a[2]	a[3]	a[4]
Value	1	5	3	2	20
Address	1000	1001	1002	1003	1004

Example

```
void fun1 (int *p, int *q){
    *p = 100;
    p = p + 2;
    *p = *q-1;
    *q = *(p+2);
    cout << "p=" << *p << " q=" << *q << "\n";
}
bool fun2 (int x[], int size) {
    int *p = x;
    int *q;
   q = p;
    for (int i=1; i<size; i++)    {
        x[i] += x[i-1];
        cout << x[i] << " ";
    }
    cout << "\n";
    fun1 (p, q);
    return (*p == *q);
}
```

```
int main ( ){
    int a[5] = {1,5,3,2,20};
    if (fun2 (a,5))
        cout << "true\n";
    else
        cout << "false\n";
    for (int i=0; i<5; i++)
        cout << a[i] << " ";
    return 0;
}
```

	main FUNCT.					fun2 FUNCT.	
Name	a[0]	a[1]	a[2]	a[3]	a[4]	**x**	**size**
Value	1	5	3	2	20		**5**
Address	1000	1001	1002	1003	1004	**2000**	**3000**

Example

```
void fun1 (int *p, int *q){
    *p = 100;
    p = p + 2;
    *p = *q-1;
    *q = *(p+2);
    cout << "p=" << *p << " q=" << *q << "\n";
}
bool fun2 (int x[], int size) {
    int *p = x;
    int *q;
   q = p;
    for (int i=1; i<size; i++)    {
        x[i] += x[i-1];
        cout << x[i] << " ";
    }
    cout << "\n";
    fun1 (p, q);
    return (*p == *q);
}
```

```
int main ( ){
    int a[5] = {1,5,3,2,20};
    if (fun2 (a,5))
        cout << "true\n";
    else
        cout << "false\n";
    for (int i=0; i<5; i++)
        cout << a[i] << " ";
    return 0;
}
```

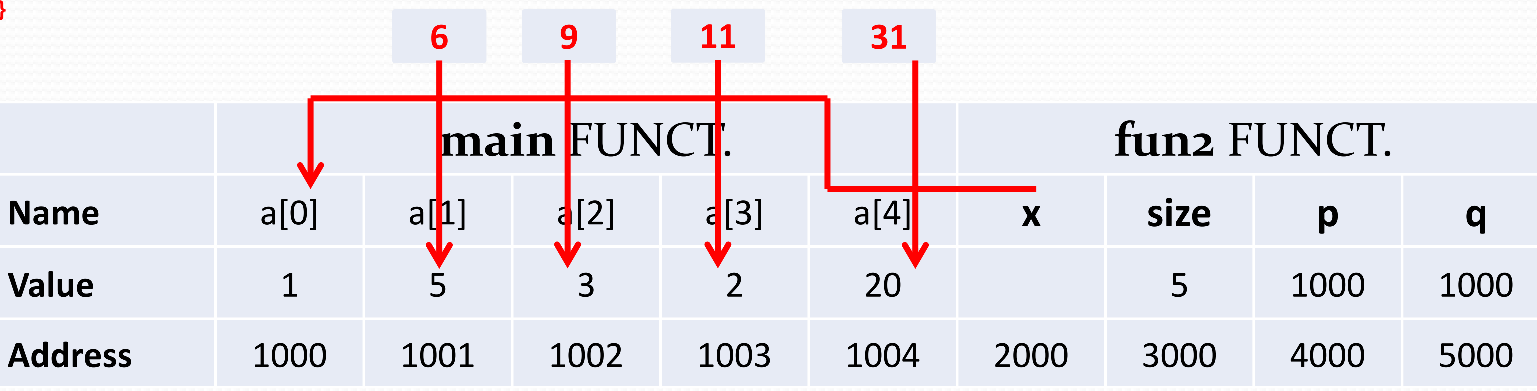
	main FUNCT.					fun2 FUNCT.			
Name	a[0]	a[1]	a[2]	a[3]	a[4]	**x**	**size**	**p**	**q**
Value	1	5	3	2	20		5	**1000**	**1000**
Address	1000	1001	1002	1003	1004	2000	3000	**4000**	**5000**

Example

```
void fun1 (int *p, int *q){
    *p = 100;
    p = p + 2;
    *p = *q-1;
    *q = *(p+2);
    cout << "p=" << *p << " q=" << *q << "\n";
}
bool fun2 (int x[], int size) {
    int *p = x;
    int *q;
   q = p;
    for (int i=1; i<size; i++)    {
        x[i] += x[i-1];
        cout << x[i] << " ";
    }
    cout << "\n";
    fun1 (p, q);
    return (*p == *q);
}
```

```
int main ( ){
    int a[5] = {1,5,3,2,20};
    if (fun2 (a,5))
        cout << "true\n";
    else
        cout << "false\n";
    for (int i=0; i<5; i++)
        cout << a[i] << " ";
    return 0;
}
```

PROGRAM OUTPUT



	main FUNCT.					fun2 FUNCT.			
Name	a[0]	a[1]	a[2]	a[3]	a[4]	**x**	**size**	**p**	**q**
Value	1	5	3	2	20		5	1000	1000
Address	1000	1001	1002	1003	1004	2000	3000	4000	5000

Example

```
void fun1 (int *p, int *q){
    *p = 100;
    p = p + 2;
    *p = *q-1;
    *q = *(p+2);
    cout << "p=" << *p << " q=" << *q << "\n";
}
bool fun2 (int x[], int size) {
    int *p = x;
    int *q;
   q = p;
    for (int i=1; i<size; i++)    {
        x[i] += x[i-1];
        cout << x[i] << " ";
    }
    cout << "\n";
    fun1 (p, q);
    return (*p == *q);
}
```

```
int main ( ){
    int a[5] = {1,5,3,2,20};
    if (fun2 (a,5))
        cout << "true\n";
    else
        cout << "false\n";
    for (int i=0; i<5; i++)
        cout << a[i] << " ";
    return 0;
}
```

PROGRAM OUTPUT

6 9 11 31

	main FUNCT.					**fun2** FUNCT.			
Name	a[0]	a[1]	a[2]	a[3]	a[4]	**x**	**size**	**p**	**q**
Value	1	6	9	11	31		5	1000	1000
Address	1000	1001	1002	1003	1004	2000	3000	4000	5000

Example

```
void fun1 (int *p, int *q){
    *p = 100;
    p = p + 2;
    *p = *q-1;
    *q = *(p+2);
    cout << "p=" << *p << " q=" << *q << "\n";
}
bool fun2 (int x[], int size) {
    int *p = x;
    int *q;
   q = p;
    for (int i=1; i<size; i++)    {
        x[i] += x[i-1];
        cout << x[i] << " ";
    }
    cout << "\n";
    fun1 (p, q);
    return (*p == *q);
}
```

```
int main ( ){
    int a[5] = {1,5,3,2,20};
    if (fun2 (a,5))
        cout << "true\n";
    else
        cout << "false\n";
    for (int i=0; i<5; i++)
        cout << a[i] << " ";
    return 0;
}
```

PROGRAM OUTPUT

6 9 11 31

	main FUNCT.					fun2 FUNCT.				fun2	
Name	a[0]	a[1]	a[2]	a[3]	a[4]	x	size	p	q	**p**	**q**
Value	1	6	9	11	31		5	1000	1000	**1000**	**1000**
Address	1000	1001	1002	1003	1004	2000	3000	4000	5000	**6000**	**7000**

Example

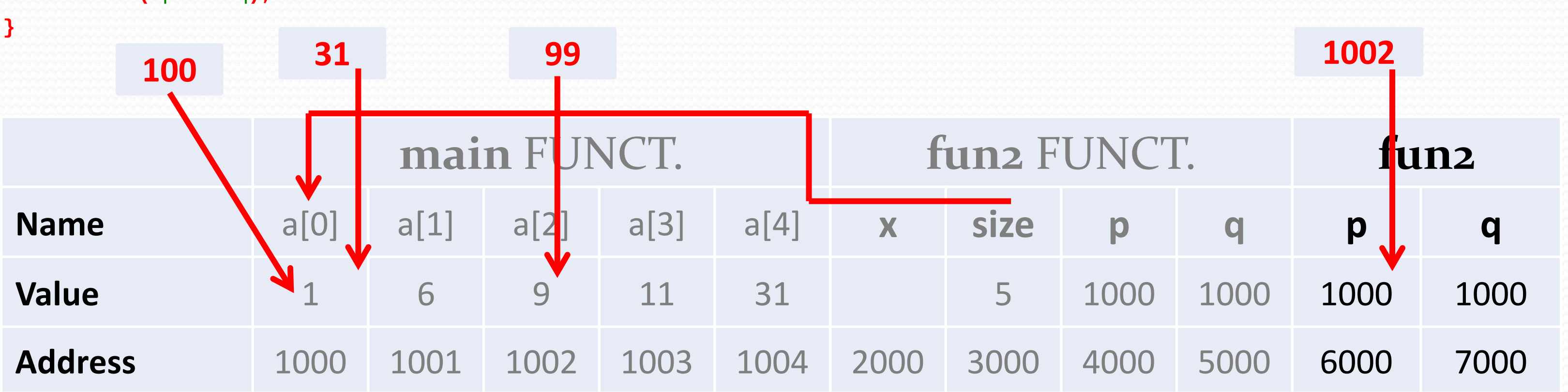
```
void fun1 (int *p, int *q){
     *p = 100;
     p = p + 2;
     *p = *q-1;
     *q = *(p+2);
     cout << "p=" << *p << " q=" << *q << "\n";
}
bool fun2 (int x[], int size) {
     int *p = x;
     int *q;
    q = p;
     for (int i=1; i<size; i++)    {
          x[i] += x[i-1];
          cout << x[i] << " ";
     }
     cout << "\n";
     fun1 (p, q);
     return (*p == *q);
}
```

```
int main ( ){
     int a[5] = {1,5,3,2,20};
     if (fun2 (a,5))
          cout << "true\n";
     else
          cout << "false\n";
     for (int i=0; i<5; i++)
          cout << a[i] << " ";
     return 0;
}
```

<u>PROGRAM OUTPUT</u>

6 9 11 31

p=99 q= 31



	main FUNCT.					fun2 FUNCT.				fun2	
Name	a[0]	a[1]	a[2]	a[3]	a[4]	x	size	p	q	p	q
Value	1	6	9	11	31		5	1000	1000	1000	1000
Address	1000	1001	1002	1003	1004	2000	3000	4000	5000	6000	7000

Example

```
void fun1 (int *p, int *q){
    *p = 100;
    p = p + 2;
    *p = *q-1;
    *q = *(p+2);
    cout << "p=" << *p << " q=" << *q << "\n";
}
bool fun2 (int x[], int size) {
    int *p = x;
    int *q;
   q = p;
    for (int i=1; i<size; i++)    {
        x[i] += x[i-1];
        cout << x[i] << " ";
    }
    cout << "\n";
    fun1 (p, q);
    return (*p == *q);
}
```

Returning value = true

```
int main ( ){
    int a[5] = {1,5,3,2,20};
    if (fun2 (a,5))
        cout << "true\n";
    else
        cout << "false\n";
    for (int i=0; i<5; i++)
        cout << a[i] << " ";
    return 0;
}
```

PROGRAM OUTPUT

6 9 11 31

p=99 q= 31

	main FUNCT.					fun2 FUNCT.			
Name	a[0]	a[1]	a[2]	a[3]	a[4]	x	size	p	q
Value	31	6	99	11	31		5	1000	1000
Address	1000	1001	1002	1003	1004	2000	3000	4000	5000

Example

```
void fun1 (int *p, int *q){
    *p = 100;
    p = p + 2;
    *p = *q-1;
    *q = *(p+2);
    cout << "p=" << *p << " q=" << *q << "\n";
}
bool fun2 (int x[], int size) {
    int *p = x;
    int *q;
   q = p;
    for (int i=1; i<size; i++)    {
        x[i] += x[i-1];
        cout << x[i] << " ";
    }
    cout << "\n";
    fun1 (p, q);
    return (*p == *q);
}
```

Returning value = true

```
int main ( ){
    int a[5] = {1,5,3,2,20};
    if (fun2 (a,5))
        cout << "true\n";
    else
        cout << "false\n";
    for (int i=0; i<5; i++)
        cout << a[i] << " ";
    return 0;
}
```

	main FUNCT.				
Name	a[0]	a[1]	a[2]	a[3]	a[4]
Value	31	6	99	11	31
Address	1000	1001	1002	1003	1004

PROGRAM OUTPUT

6 9 11 31

p=99 q= 31

true

Example

Example

- We have an array with numbers stored in it. Write a program that cleans recurrent numbers in the array and finally creates a new array with the unique numbers and repetition times in the original array.
- Let's say the array has the elements:
  - {12,32,-9,0,34,12,33,-9,-9,4,12,32,12,43,44,12,-9,0,44,43};
  - We should define a new array of same length
    - Since the max length of the new cleaned version is same as the length of the original one

Example

```
int arr[]= {12,32,-9,0,34,12,33,-9,-
9,4,12,32,12,43,44,12,-9,0,44,43};
//Number of elements is found
int N=sizeof(arr)/sizeof(arr[0]);

//Cleaned version
int arrNew[N];

//Numbers of repetitions
int numbers[N];

//First elements are same
int sizeNew=0;
numbers[sizeNew]=1;
arrNew[sizeNew++]=arr[0];
```

arr	arrNew	numbers	sizeNew
12	12	1	1
32			
34			
12			
33			
-9			
-9			
4			
12			
32			
12			
43			
44			
12			
-9			
0			

Example

- We should compare each element of arr with arrNew.
- If we have the element in arrNew then increase the repetition times by one.
- If it is NOT in the arrNew, then add this element to arrNew
- Set repetition time to 1
- and increase the size of arrNew by 1.

arr	arrNew	numbers	sizeNew
12	12	1	1
32			
34			
12			
33			
-9			
-9			
4			
12			
32			
12			
43			
44			
12			
-9			
0			

Example

- Start with 32 (arr[1])
- Not in the list of arrNew, then add it to the end.
- Set the repetition number for the corresponding index to 1.
- Increase the sizeNew by 1.

arr	arrNew	numbers	sizeNew
12	12	1	2
32	32	1	
34			
12			
33			
-9			
-9			
4			
12			
32			
12			
43			
44			
12			
-9			
0			

Example

- Go to the next element : 34 (arr[2])
- Compare it to arrNew[0]
  - false
- Compare it to arrNew[1]
  - false
- Not in the list of arrNew, then add it to the end.
- Set the repetition number for the corresponding index to 1.
- Increase the sizeNew by 1.

arr	arrNew	numbers	sizeNew
12	12	1	3
32	32	1	
34	34	1	
12			
33			
-9			
-9			
4			
12			
32			
12			
43			
44			
12			
-9			
0			

Example

- Go to the next element : 12 (arr[3])
- Compare it to arrNew[0]
  - true
- Increase the repetition times for the corresponding index by 1.
- Do not increase sizeNew.

arr	arrNew	numbers	sizeNew
12	12	2	3
32	32	1	
34	34	1	
12			
33			
-9			
-9			
4			
12			
32			
12			
43			
44			
12			
-9			
0			

Example

- We should repeat the process for every element in arr.

```
int j;
//Loop for elements in arr
for (int i=1; i<N; i++)
{
    //Loop to compare with arrNew
    for(j=0; j<sizeNew; j++)
    {   //If it is in arrNew
        if(arr[i]==arrNew[j])
        {   //increase the repetition time
            numbers[j]++;
            //don't have to look for other elements in arrNew
            //since they are unique
            break;
        }
    }
    //Did we find arr[i] in arrNew
    if(j==sizeNew)
    {   //If not add the element to the end of arrNew
        numbers[sizeNew]=1;
        arrNew[sizeNew++]=arr[i];
    }
}
```

Example

- Finally print out the result

```
for(j=0; j<sizeNew; j++)
        cout<<arrNew[j]<<"\t «
            <<numbers[j]<<endl;
```

```
12          5
32          2
34          1
33          1
-9          3
4           1
43          1
44          1
0           1
```

arr	arrNew	numbers	sizeNew
12	12	5	9
32	32	2	
34	34	1	
12	33	1	
33	-9	3	
-9	4	1	
-9	43	1	
4	44	1	
12	0	1	
32			
12			
43			
44			
12			
-9			
0			


Q&A